BenchCouncil Transactions on Benchmarks, Standards and Evaluations 2 (2022) 100037

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: https://www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Training, testing and benchmarking medical AI models using Clinical AIBench

Yunyou Huang [a,f], Xiuxia Miao [a], Ruchang Zhang [a], Li Ma [c], Wenjing Liu [a], Fan Zhang [b], Xianglong Guan [a], Xiaoshuang Liang [a], Xiangjiang Lu [a], Suqing Tang [e], Zhifei Zhang [d,*]

[a] Guangxi Key Lab of Multi-Source Information Mining & Security, School of Computer Science and Engineering & School of Software, Guangxi Normal University, Guilin, China
[b] State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[c] Guilin Medical University, Guilin, China
[d] Department of Physiology and Pathophysiology, Capital Medical University, Beijing, China
[e] Faculty of Education, Guangxi Normal University, Guilin, China
[f] International Open Benchmark Council, Beijing, China

ARTICLE INFO

Keywords:
Benchmark
Clinical setting
Alzheimer's disease
COVID-19
Dental
Configurable clinical setting

ABSTRACT

AI technology has been used in many clinical research fields, but most AI technologies are difficult to land in real-world clinical settings. In most current clinical AI research settings, the diagnosis task is to identify different types of diseases among the given ones. However, the diagnosis in real-world settings needs dynamically developing inspection strategies based on the existing resources of medical institutions and identifying different kinds of diseases out of many possibilities. To promote the development of different clinical AI technologies and the implementation of clinical applications, we propose a benchmark named Clinical AIBench for developing, verifying, and evaluating clinical AI technologies in real-world clinical settings. Specifically, Clinical AIBench can be used for: (1) Model training and testing: Researchers can use the data to train and test their models. (2)Model evaluation: Researchers can use Clinical AIBench to objectively, fairly, and comparably evaluate various models of different researchers. (3) Clinical value evaluation: Researchers can use the clinical indicators provided by Clinical AIBench to evaluate the clinical value of models, which will be applied in real-world clinical settings. For convenience, Clinical AIBench provides three different levels of clinical settings: restricted clinical setting, which is named closed clinical setting, data island clinical setting, and real-world clinical setting, which is called open clinical setting. In addition, Clinical AIBench covers three diseases: Alzheimer's disease, COVID-19, and dental. Clinical AIBench provides python APIs to researchers. The data and source code are publicly available from the project website https://www.benchcouncil.org/clinical_aibench/.

1. Introduction

Data is the cornerstone of current AI technologies research, which has become the consensus of researchers. From ImageNet [1] and MNIST [2] in the general field to ADNI [3] and TCGA [4] in the clinical field, the datasets have greatly promoted the development of AI technologies research. However, many current AI technologies proposed by research work are difficult to land in the real world. Especially in clinical medicine, only 71 clinical AI technologies have been approved by the FDA by now [5]. An AI-assisted decision-making device has been used to detect the fetus's heart rate during childbirth in pregnant women. Still, the practice proved that the device could not improve the clinical outcome of the mother or baby; Instead, it increases the workload of clinicians [6].

Why the AI technology loses its power in the real world? The answer is that the essentials of the same clinical tasks are changed in different clinical settings. For example, the diagnosis tasks in most current clinical AI research settings are identifying different types of diseases among the given diseases. The diagnosis in the isolated data island clinical setting, also called federated learning setting [7], has three phases: first, ensure data security; second, use the data from different institutions to train a model; finally, identify different types of diseases among the given ones. Constantly, the diagnosis in real-world settings, which we also call open clinical settings, needs dynamically developing inspection strategies based on the existing resources of medical institutions, locating subjects in one type or several types of diseases, and identifying different kinds of diseases. Thus, solving a task in a clinical setting, e.g., a closed or data island clinical setting, does not mean it is well done in an open clinical setting.

To promote the clinical AI technology implementation, we propose a scenario-based benchmark suite [8], named Clinical AIBench, which

* Corresponding author.
E-mail address: zhifeiz@ccmu.edu.cn (Z. Zhang).

https://doi.org/10.1016/j.tbench.2022.100037
Received 7 March 2022; Received in revised form 24 March 2022; Accepted 24 March 2022
Available online 12 April 2022

is a measurement standard and also requires rigorous validation [9], to develop, verify, and evaluate the clinical AI technologies and systems [10]. In detail, Clinical AIBench can be used for (1) model training and testing. For example, the closed clinical setting mentioned in Section 5.1, the data island clinical setting mentioned in Section 5.2, and the open clinical setting mentioned in Section 5.3 can be used for training and testing Alzheimer's disease diagnosis models under different requirements. (2) Model evaluation. For example, the open clinical setting mentioned in Section 5.3 reproduces the clinical setting's main features of the real Alzheimer's disease diagnosis, not allowed to be modified. Not only does it can be used to train and test Alzheimer's disease diagnosis model, but it also can be used as a fair, objective, and comparable tool to evaluate various models from different researchers. (3) Clinical value evaluation. For example, the **get_damage_level** and **get_cost** APIs mentioned in Section 4 can be used to evaluate the cost and the degree of damage of the subjects during the diagnosis. The risks indicators and the subject's benefits during diagnosis are an integrated part of Clinical AIBench.

2. The installation of Clinical AIBench

Clinical AIBench is an open-source tool and provides services according to the Client-Services model (currently, Clinical AIBench only runs on Linux and Mac systems). Clinical AIBench is designed to be similar to the FLBench [7] structure, which has four parts: input data, scenario configuration, scenario, and automated deployment tools. The researcher can download the clients of Clinical AIBench from this website. Due to the particularity of clinical data and the requirements of the organization of data owners, researchers must register and follow all relevant protocols.

As shown in Fig. 1, Clinical AIBench requires 13 packages: configparser, pandas, numpy, nibabel, flask, scikit-learn, requests, imageio, joblib, tensorflow, torch, wandb, fedml [11] etc.

3. The configuration of clinical setting

In Clinical AIBench, the clinical setting is constructed according to the configuration file. For every typical clinical setting, Clinical AIBench will provide a configuration file to describe the characteristic of the clinical setting and build the clinical setting in Clinical AIBench. In addition, the researcher is able to upload the configuration file to construct their clinical setting according to their needs. As shown in Fig. 2, the format of the configuration file used in Clinical AIBench is *ini*. The *ini* file has three elements: section, parameter, and comment. The *section* is enclosed in a square brackets(*[]*), and an *ini* file contains one or more section. The *parameter* is a key–value pair. All parameters after the section declaration belong to the section. The *comment* starts with *;* or *#*, which means the whole line is a comment. As shown in Fig. 3 is a specific example of the use of *ini* file.

As shown in Fig. 3, this configuration file has two sections. Each section contains multiple parameters. The first section, named *scenario*, describes some basic information about the *scenario*, such as the *scenario* name, the *scenario* version number, etc. The second section, named *block1*, describes the data information used in the *scenario*. The meaning of parameters of the *block1* will be described in detail below.

(1) *files* specifies the data files that need to be used in the scenario. The requirements for the data file are as follows: (1) The file type must be *csv* format. (2) The first line of the file must be the column names. (3) Special symbols cannot exist in column names, such as @, ., & etc. (4) The path of image, audio, or video will be filled in the file if the data file contains the image, audio, or video resource.
(2) *joinorder* specifies the order that connect data files, such as *joinorder = file1, file2.*

```
configparser==5.0.2
pandas==1.0.5
numpy==1.18.5
nibabel==3.2.1
flask==1.1.2
scikit-learn==0.23.1
requests==2.24.0
imageio==2.9.0
joblib==1.0.1
tensorflow==2.6.0rc0
torch==1.9.1
wandb==0.12.2
fedml==0.1
```

Fig. 1. Packages.

```
[section1]
key1=value1, value2, ...
key2=value1, value2, ...

[section2]
key1=value1, value2, ...
key2=value1, value2, ...

```

Fig. 2. Ini format.

(3) *connection* specifies the primary keys that connect different data files, such as *connection = file1|column1&file2|column1+ file1|column2&file2|column2.*
(4) *columns* specifies the column that selected in the file according to column name, such as *columns = files1|column1, files1|column2, files2|column1, files2|column2.* And all data in the file is selected if the column name is no designated.
(5) *rows* specifies the row selection rule of the file according to the specified conditions, such as *rows = file1|column1 = value1, file1|column2! = value2.* And all data in the file is selected if the *row* is no designated.
(6) *label* specifies which column of the file as the label of machine learning task, such as *label = file| column1.*
(7) *partition_number* specifies the number of clients. If the number is 1, which is a closed clinical setting. If the number is greater than 1, which is a data island clinical setting with multiple clients. If the number is −1, which is a data island clinical setting with an indeterminate number of clients.
(8) *assignment* specifies the method that divide datasets in the data island clinical setting: (1) *_scope* the function of this method is divide datasets according to sorted of the datasets and the parameter *rate* in the configuration file; (2) *_class* the function of this method is divide the datasets according to the type of value that is designated column; (3) *_constraint* the function of this method is divide the datasets according to the parameter *condition* in the configuration file.

```
[scenarion]

name
version
author

[block1]

files

joinorder

connection

columns

rows

label

partition_number

;assignment=scope()
;assignment=class()
;assignment=constraint()

;rate

;conditions

;add

image

audio

video
```

Fig. 3. Configuration file.

(9) *rate* specifies the proportion of partition data, which allocate to each client in the data island clinical setting. The number of proportions needs to be equal to the number of clients.
(10) *conditions* specifies the condition that select the data in the data island clinical setting, such as *conditions = file1|column1> value1&file1|column1< value2, file1|column1 = value3*.
(11) *add* specifies the fine adjustment method of the data, which has been partitioned, such as *add = partition_number@file1|column1> conditions*.
(12) *image* specifies the column, which is image data in the file, such as *image = file|column*.
(12) *audio* specifies the column, which is audio data in the file, such as *audio = file|column*.
(12) *video* specifies the column, which is video data in the file, such as *video = file|column*.

4. The main API of Clinical AIBench

Clinical AIBench provides many python APIs for researchers to utilize the Clinical AIBench to train, test, and evaluate their AI models.

(1) *loadData(inipath, data_save_path, proportion = [80, 5, 15], setting = −1)*: The function of this method is loading the training set, validation set and test set from the local disk. Each parameter is defined as follows.

  (1) *inipath* specifies the path of the configuration file.
  (2) *data_save_path* specifies the saving path of data.
  (3) *proportion* specifies the partition proportion for the training set, validation set, and test set, and the default partition proportion is: 80% for the training set, 5% for the validation set, and 15% for the test set.
  (4) *setting* specifies the scenario type. It is the closed clinical setting or the data island clinical setting if the *setting* is −1. It is the open setting clinical if the *setting* is −2.

(2) *_toMatrix(source, save_path)*: The function of this method is obtaining feature matrix X and label matrix Y. Each parameter is defined as follows.

  (1) *source* specifies the data that has been downloaded.
  (2) *save_path* specifies the saving path that are the feature matrix X and the label matrix Y.

(3) *feature_extraction(input_path, image_tmp_save_path, image_save_path, pre_model = "DenseNet201", pooling = "avg", start = −1, end = −1)*: The function of this method is extracting features from a registered image by the pre-trained model. Each parameter is defined as follows.

  (1) *input_path* specifies the path of the image that needs extracting feature.
  (2) *image_tmp_save_path* specifies the temporary saving path of the image.
  (3) *image_save_path* specifies the saving path of the image, which was extracted feature.
  (4) *pre_model* specifies the pre-trained model.
  (5) *pooling* specifies the pooling method.
  (6) *start* and *end* specifies the identifier that respectively is the start and end position of the number of images.

(4) *convert_3Dto2DtoRGB(image_path, image_tmp_save_path, image_save_path)*: The function of this method is converting 3D image data to 2D image data, and then converting the gray image to RGB image. Each parameter is defined as follows.

  (1) *image_path* specifies the path of the image that needs to be converted.
  (2) *image_tmp_save_path* specifies the temporary saving path of the image.
  (3) *image_save_path* specifies the saving path of image, which was converted.

(5) *registration(input_path, out_path, image_type, weighted_image)*: The function of this method is aligning a image to a common template. Each parameter is defined as follows.

  (1) *image_path* specifies the path of the image that needs to be registered.
  (2) *output_path* specifies the saving path of the image, which was registered.
  (3) *image_type* specifies the type of the image.
  (4) *weighted_image* specifies the type of MRI-weighted image.

(6) *image_correct(image_path, image_save_path, image_type)*: The function of this method is correcting the non-uniform tissue intensity image into an image with uniform tissue intensity. Each parameter is defined as follows.

  (1) *image_path* specifies the path of the image that needs to be corrected.
  (2) *image_save_path* specifies the saving path of the image, which was corrected.
  (3) *image_type* specifies the type of the image.

(7) *strip_nonbrain(image_path, image_save_path, strip_method)*: The function of this method is stripping the non-brain structure components, such as the skull, neck, scalp and so on. Each parameter is defined as follows.

  (1) *image_path* specifies the path of the image that needs to be stripped.
  (2) *image_save_path* specifies the saving path of the image, which was stripped.
  (3) *strip_method* specifies the stripping method.

(8) *image_segmentation(image_path, image_save_path, segment_method)*: The function of this method is segmenting the image of pre-processed into different tissue types according to the different brain components. Each parameter is defined as follows.

(1) *image_path* specifies the path of the image that needs to be segmented.
(2) *image_save_path* specifies the saving path of the image, which was segmented.
(3) *segment_method* specifies the segmentation method.

(9) *smoothing(image_path, image_save_path, smooth_method)*: The function of this method is compensating the inaccuracy of registration by reducing the image noise and making the image blur. Each parameter is defined as follows.

(1) *image_path* specifies the path of the image that needs to be smoothed.
(2) *image_save_path* specifies the saving path of the image, which was smoothed.
(3) *smooth_method* specifies the smoothing method.

(10) *get_damage_level(method, predict_value, label)*: The function of this method is obtaining the damage value that the subject suffered during the medical diagnosis and treatment. These damage levels are scale from 0 to 5, and the 5 level is the highest. Each parameter is defined as follows.

(1) *method* specifies the examination items of the subject during the diagnosis and treatment.
(2) *predict_value* specifies the predicted value of the model. The predicted value is the disease type if it is a disease diagnosis prediction. The predicted value is the next treatment if it is a treatment prediction.
(3) *label* specifies the actual disease type of the subject or the next treatment method advised by clinicians.

(11) *get_cost(method, predict_value, label)*: The function of this method is obtaining the cost of the subjects during the medical diagnosis and treatment. Each parameter is defined as follows.

(1) *method* specifies the examination items of the subject during the diagnosis and treatment.
(2) *predict_value* specifies the predicted value of the model. The predicted value is the disease type if it is a disease diagnosis prediction. The predicted value is the next treatment if it is a treatment prediction.
(3) *label* specifies the actual disease type of the subject or the next treatment method advised by clinicians.

5. The Clinical AIBench use cases

In order to demonstrate the usage of Clinical AIBench, the diagnosis model of Alzheimer's disease is developed, verified, and evaluated in three different clinical settings, respectively.

5.1. Closed clinical setting

Currently, most clinical AI researches are based on the closed clinical setting: (1) The types of all subjects are already known, that is, the types of all subjects appear in the training set. (2) The data types of all subjects are the same. That is, all subjects contain all pre-specified types of data. (3) All medical institutions can obtain all pre-specified types of data, that is, all medical institutions have pre-specified capabilities of examination and treatment.

In this paper, according to the characteristics of the restricted scenario, we construct a closed clinical setting based on the basic information of Alzheimer's disease subjects and MRI images: (1) The clinical setting contains 2127 subjects, of which 740 are Alzheimer's disease (AD), 1082 are mild cognitive impairment (MCI), and 589 are cognitively normal subjects (CN). (2) All types of subjects are divided into training set, verification set, and test set at a ratio of 80%:5%:15%. (3) Because subjects generally have multiple follow-up visits, Clinical AIBench stipulates that the data of different visits of the same subject is only allowed to appear in one of the training set, the verification set, and the test set. (4) All subjects contain basic information and MRI information. When a subject is missing some part of the data, Clinical AIBench will fill in the data according to a specified strategy.

To construct the clinical setting above, we create a configuration file as shown in Fig. 4.

In this paper, we develop and evaluate a diagnosis model of Alzheimer's disease in above clinical setting. As shown in Fig. 5, it is a model of Alzheimer's classification based on Keras. As shown in Fig. 6, it is the entry point of the program. Researchers can start the *main()* function to execute the entire program. Line 21 of Fig. 6 is to create a model object. Line 34 of Fig. 6 is the model evaluation. Line 39 of Fig. 6 is the model prediction.

5.2. Data island clinical setting

Due to data privacy, data security, and data value, clinical data is difficult to share for researchers. The clinical data is usually stored in their medical institution, forming the data island clinical setting. In order to promote the progress of AI technologies in the data island clinical setting, new machine learning concepts (federated learning and swarm learning) are proposed [12–14].

In this paper, we construct a data island clinical setting: (1) The subject and data types are similar to the closed clinical setting. (2) Divide the clinical data into multiple parts according to different medical institutions to form a natural data island clinical setting.

To construct the clinical setting above, we create a configuration file as shown in Fig. 7. The difference between this scenario configuration file and the above closed scenario configuration file is that the number of clients is sets to −1 (in line 23 of Fig. 7), and the data partition method selected the *_class* (in line 26 of Fig. 7).

In this paper, we develop and evaluate a diagnosis model of Alzheimer's disease in above clinical setting. As shown in Fig. 8, it was a model of Alzheimer's classification based on Pytorch. As shown in Fig. 9, it is the entry point of the program. The 46 line of Fig. 9 is the entrance to the model training [11].

5.3. Open clinical setting

As mentioned above, the real-world clinical setting is open and full of complexity and uncertainty. We capture the characteristics of the real-world clinical setting and construct the open clinical setting which maintains the main features of the real-world clinical setting: (1) The type of some subjects is unknown, that is, some type of subject in the test set does not appear in the training set and verification set. (2) The specific conditions of the subjects are different, and the examinations and treatment methods executed on the subjects are different, that is, the data types of the subjects are different. (3) The medical conditions of medical institutions are different. Some institutions can meet the requirements of the diagnosis and treatment strategies of subjects, while others cannot. That is to say, the types of data of subjects collected in different medical institutions are many different.

In this paper, we do not provide the modifiable function of the open clinical setting. And we develop and evaluate a diagnosis model of Alzheimer's disease in the above clinical setting. Clinical AIBench proposes a unified data representation framework for the open clinical setting of Alzheimer's disease, since the different dimensions of each data category, the number of data categories included in each visit are different, and the number of history visits included in each subject is also different. The data framework presents an examination category

```
[scenarion]

name= Closed clinical setting
version=v1.0
author=TEST

[block1]

files= merger@../data/ADNIMERGE_without_bad_value.csv, image@../data/image_information.csv

joinorder=merger, image

connection=merger|RID&image|RID+merger|VISCODE&image|Visit,

columns = merger|RID , merger|VISCODE , merger|SITE ,  merger|DX , merger|PTGENDER ,merger|APOE4,
  merger|PTEDUCAT , merger|PTETHCAT , merger|PTRACCAT , merger|PTMARRY , image|RID , image|Visit , image|Modality ,
image|Sequence, image|SavePath


rows =  image|Modality = MRI , image|Sequence = 1, merger|DX != null

label= merger|DX

partition_number = 1

;assignment=scope(sort_by=merger|SITE, order=desc)
;assignment=class(group=merger|SITE)
;assignment=constraint()

;rate=0.1,0.5,0.1,0.1,0.2

;conditions=merger|RID>2 & merger|RID<10,  merger|RID<100, merger|SITE=11, merger|SITE=100 & merger|RID>50,
merger|EXAMDATE > 2018/5/15

;add=1@gene|EXAMDATE>2018/5/15@1 , 5@gene|SITE=11@0.85

image=image|SavePath

audio=

video=
```

Fig. 4. Closed clinical setting configuration file.

```
class Closed_clinical_model(tf.keras.Model):
    def __init__(self, num_classes=3):
        super(Closed_clinical_model, self).__init__()
        self.num_classes = num_classes
        self.layer1 = layers.Dense(256, activation='relu')
        self.layer2 = layers.Dense(128, activation='relu')
        self.layer3 = layers.Dense(64, activation='relu')
        self.layer4 = layers.Dense(32, activation='relu')
        self.layer5 = layers.Dense(num_classes, activation='relu')

    def call(self, inputs):
        h1 = self.layer1(inputs)
        h2 = self.layer2(h1)
        h3 = self.layer3(h2)
        h4 = self.layer4(h3)
        out = self.layer5(h4)
        return out
```

Fig. 5. Closed clinical setting model.

in the subject's visit by an array with a shape of 1×2090. The shape of our data is $n \times 2090$, n is the number of categories of data for the subject. The data of the open clinical setting of Alzheimer's disease is able downloaded by Clinical AIBench. And as a sample, we develop a model based on the open clinical setting of Alzheimer's disease shown in the file OpenClinicalAI.py and HierarchicalOpenNet.py [15].

6. The clinical data in the Clinical AIBench

Clinical AIBench is an open and evolving benchmark. Datasets that we select will follow the following principles (1): diseases that are more concerned in the current clinical research field; (2): datasets that are more complete; (3): and datasets that are low acquisition cost. Currently, it contains three clinical datasets: Alzheimer's disease (more complete), COVID-19 (more concerned), and dental (low acquisition cost). In addition, the ICU and psychiatric disorders datasets will be added to Clinical AIBench soon.

6.1. Privacy and data security

The clinical data is very sensitive, involving subject privacy and data security. According to the security objectives in Section 5 of the

```python
if __name__ == '__main__':


    # 1.configuration file path
    iniPath = '/home/ini/dataset.ini'

     # 2.data set save path
    dataSavePath = '/home/downloadData'

    # 3.create scenario  configuration object
    client = SCClient()

    # 4. load the data set according to the configuration file (if it is downloaded, load it directly, if it has
    # not been downloaded, download it first, and then load it back)
    # proportion: the proportion of training set, validation set and test set. The default division ratio is: 80%
    # for training set, 50% for validation set, and 15% for test set.
    X_train, y_train, X_verif, y_verif, X_test, y_test = client.loadData(iniPath,
                                                                         dataSavePath,
                                                                         proportion=[80, 5, 15],)


    # create a model object
    model = Closed_clinical_model(num_classes=3)

    # model assembly
    model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.01),
                  loss=tf.keras.losses.CategoricalCrossentropy(from_logits=True),
                  metrics=['accuracy']
                  )

    # model training
    model.fit(X_train, y_train, batch_size=32, epochs=200, validation_data=(X_verif, y_verif),
                          validation_freq=2)

    # model evaluation
    loss, accuracy = model.evaluate(X_test, y_test, batch_size=64)
    print("accuracy:")
    print(accuracy)

    # model prediction
    pred = model.predict(X_test)

    print("pred:")
    print(pred)

    print("y_test:")
    print(y_test)
```

Fig. 6. Closed clinical setting's main function.

standard GB/T 39725-2020 "Information security technology-Guide for health data security" [16], researchers need to ensure the confidentiality, integrity, and availability of clinical data and ensure that the usage of the clinical data is legal. In addition, in accordance with the standard GB/T 39725-2020 "Information security technology-Guide for health data security" Section 7 as mentioned in point 4 of point b, when restricted datasets are used for scientific research, medical/health education, and public health purposes, the corresponding personal health, and medical data can be used without the authorization of the subject. Thus the public data of Clinical AIBench currently only provides services for scientific research.

The private data of Clinical AIBench will be strictly desensitized according to the Health Insurance Portability and Accountability Act/1996, Public Law 104–191 and technical standards. In addition, all private data is stored in the private server of the data owner, and provides services through Clinical AIBench with the data island clinical setting manner.

6.2. Alzheimer's disease

The data support of Alzheimer's disease clinical setting is derived from the ADNI dataset, as shown in Table 1, which contains 9591 individual visits of 2127 subjects [17]. We organize the ADNI dataset according to the subject identification (RID) and the visit code (VISCODE2, which represents the time between the current visit date of subjects and the first visit date). The dataset is divided into the following parts.

(1) The merge table has 113 fields, including the classification of subjects and the information currently generally considered to be related to Alzheimer's disease, and the value corresponding to the first visit of that information. Full merge table information is available at the website Merge table, where for unified expression and understanding field VISCODE is expressed as VISCODE2.
(2) The basic information table has 148 fields, including the subject's demographic information, family medical history, personal medical history, and the subject's current symptoms. More information is available at Basic information table.
(3) The physical examination information table contains 39 fields, including the subject's neurological examination information, physical examination information, and vital signs information. More information is available at Physical examination information table.
(4) The cognitive information table has 259 fields, including Alzheimer's Disease Assessment Scale-Cognitive information (ADASCog), Mini Mental State Exam information (MMSE), Montreal Cognitive Assessment information (MoCA), Clinical Dementia Rating information (CDR), and Cognitive Change Index information (CCI). More information is available at Cognitive information table.
(5) The cognitive test information table has 73 fields, including neuropsychological information: Clock Painting Test, Logic Memory Test-I (Instant Memory), Logic Memory Test-II (Time-lapse Memory), Rey Hearing Test, Wired Test, Animal Category Test, Boston Naming Test, American Adult Reading Test. More information is available at Cognitive test information table.
(6) The function and behavior test table has 112 fields, including Functional Assessment Questionnaire information (FAQ), Everyday Cognition — Participant Self-Report information

```
[scenarion]

name= Data island clinical setting
version=v1.0
author=TEST

[block1]

files= merger@../data/ADNIMERGE_without_bad_value.csv, image@../data/image_information.csv

joinorder=merger, image

connection=merger|RID&image|RID+merger|VISCODE&image|Visit,

columns = merger|RID , merger|VISCODE , merger|SITE ,  merger|DX , merger|PTGENDER ,merger|APOE4,
  merger|PTEDUCAT , merger|PTETHCAT , merger|PTRACCAT , merger|PTMARRY , image|RID , image|Visit , image|Modality ,
image|Sequence, image|SavePath


rows =  image|Modality = MRI , image|Sequence = 1, merger|DX != null

label= merger|DX

partition_number = -1

;assignment=scope(sort_by=merger|SITE, order=desc)
assignment=class(group=merger|SITE)
;assignment=constraint()

rate=0.1,0.5,0.1,0.1,0.2

;conditions=merger|RID>2 & merger|RID<10,  merger|RID<100, merger|SITE=11, merger|SITE=100 & merger|RID>50,
merger|EXAMDATE > 2018/5/15

;add=1@gene|EXAMDATE>2018/5/15@1 , 5@gene|SITE=11@0.85

image=image|SavePath

audio=

video=
```

Fig. 7. Data island clinical setting configuration file.

```python
class Data_island_clinical_model(nn.Module):
    def __init__(self, num_classes=3):
        super(Data_island_clinical_model, self).__init__()
        self.fc1 = nn.Linear(1930, 200)
        self.fc2 = nn.Linear(200, 200)
        self.fc3 = nn.Linear(200, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = x.view(x.shape[0], -1)
        x = self.relu(self.fc1(x))
        x = self.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Fig. 8. Data island clinical setting model.

(ECOGPT), and Everyday Cognition — Study Partner Report information (ECOGSP). More information is available at Function and behavior table.

(7) The psychiatric test table has 55 fields, including the subject's Geriatric Depression Scale information (GDS) and Neuropsychiatric Inventory Examination information (NPI). More information is available at Psychiatric test table.

(8) The blood table has 459 fields, including the subject's various blood test information. More information is available at Blood table.

(9) The urine table has 4 fields, including the subject's various urine test information. More information is available at Urine table.

(10) The cerebral spinal fluid (CSF) table has 379 fields, including the subject's various cerebrospinal fluid test informations. More information is available at Cerebral spinal fluid table.

```python
if __name__ == '__main__':

    # 1.configuration file path
    iniPath = '/home/ini/dataset_FL.ini'

    # 2.data set save path
    dataSavePath = '/home/downloadData_FL'

    # 3.create scenario configuration object
    client = SCClient()

    # 4.load the data set according to the configuration file (if it is downloaded, load it directly, if it has
    # not been downloaded, download it first, and then load it back)
    # according to different sites, the data set is divided into different subsets.
    # proportion: The first two digits represent the start and end of the site.
    # the last three digits represent the division ratio of the training set, validation set
    # and test set of the corresponding site data set.
    X_train, y_train, X_verif, y_verif, X_test, y_test = client.loadDataTest1(iniPath,
                                                                              dataSavePath,
                                                                              proportion=[[[5, 10], 85, 0, 15],
                                                                                          [[20, 10], 70, 0, 30]],)
    # federated learning:

    # create training parameter object
    parser = add_args(argparse.ArgumentParser(description='FedAvg-standalone'))
    args = parser.parse_args()
    device = torch.device("cuda:" + str(args.gpu) if torch.cuda.is_available() else "cpu")

    # call visualization function
    _wandb(args, device)

    # load the divided data set
    dataset = load_partition_data_adni(X_train, y_train, X_test, y_test, train_bs=64,
                                       test_bs=64)

    # create model object
    model = Data_island_clinical_model(num_classes=dataset[7])

    # call custom model trainer
    model_trainer = main_feavg.custom_model_trainer(args, model)

    logging.info(model)

    fedavgAPI = fedavg_api.FedAvgAPI(dataset, device, args, model_trainer)

    # federal Learning Training
    fedavgAPI.train()
```

Fig. 9. Data island clinical setting's main function.

(11) The gene table has 351 fields, including subject's single nucleotide polymorphisms (SNPs) related to Alzheimer's disease. Only one genetic test was performed per subject in Clinical AIBench, so the genetic test table does not contain visit codes and defaults to the data at the first visit. In addition, each SNP is represented by 5 fields, which are the four bases A, G, C, T, and the SNP's confidence. More information is available at Gene table.

(12) The medical imaging table has 33 fields, including the subject's 3 types of original medical images and information: MRI, PET-18FDG, PET-AV45. More information is available at Medical imaging table.

6.3. COVID-19

The COVID-19 data support in Clinical AIBench is collected from different publicly accessible datasets, online resources, and published papers. The COVID-19 dataset contains three different publicly available datasets: covid-chestxray-dataset [18], Figure1-COVID-chestxray-dataset [19], and Actualmed-COVID-chestxray-dataset [20].

As shown in Table 2, which contains 1243 individual visit records of 742 subjects. The summary table of COVID-19 has 24 fields, including the demographics of subject, current physical condition of subject, original medical image, and image type. More information is available at COVID-19 table.

6.4. Dental

The dental dataset is a private dataset from cooperative medical institutions that was desensitized. The dental dataset can only be used to develop, validate, and evaluate clinical AI models or systems similar to the data island clinical setting, and runs only on the private server of the data owner.

As shown in Table 3, the dental dataset contains 661 individual visits of 447 subjects. The dental data table has 5 fields, including the id, data of visit, age, gender, saving path of image.

7. Precautions for using Clinical AIBench

Clinical AIBench can be accessed according to the standard data usage agreement (different clinical settings involve different usage agreements). According to this agreement, users must agree to only use Clinical AIBench for the purposes stated in the agreement. The users of Clinical AIBench should thank the original authors and research laboratories for their contributions by correctly citing related article and their links to Clinical AIBench.

To objectively, fairly, and impartially evaluate the clinical AI models and systems of researchers. Clinical AIBench as an independent third party tool, which will provide a consistent clinical setting for evaluation. The configuration file of Clinical AIBench provided clinical setting is not allowed to be modified.

The data of Clinical AIBench comes from two sources: public datasets and private datasets. The public datasets can be downloaded for the development, verification, and evaluation of clinical models or systems of researchers. However, the private datasets can only be used to develop, validate, and evaluate clinical AI models or systems similar to the data island clinical setting; that is, datasets are stored on the private server of the data owner and cannot be downloaded.

Table 1
Characteristics of Alzheimer's disease subjects.

		Subjects
Age	54–59.9	80
	60–69.9	596
	70–70.9	1048
	80–80.9	395
	90–91.9	6
	Unknow	2
Gender	Female	1130
	Male	997
Educate	4–7	11
	8–10	40
	11–13	353
	14–16	823
	17–20	900
Ethnic category	Hisp/Latino	73
	Not Hisp/Latino	2042
	Unknown	12
Racial category	Asian	40
	Black	88
	Hawaiian/Other PI	2
	More than one	25
	White	1964
	Am Indian/Alaskan	4
	Unknown	4
Marriage	Married	1618
	Never_married	73
	Widowed	238
	Divorced	191
	Unknown	7
Category	AD	740
	CN	589
	MCI	1082
	SMC	280

Table 2
Summary of COVID-19 subjects.

		Subjects
Age	18–30	48
	31–50	122
	51–70	134
	71–94	82
	Unknown	356
Sex	Female	169
	Male	281
	Unknown	292
Offset	0–20	312
	21–40	10
	41–60	4
	<0	10
	61–365	3
	Unknown	403
Category	No finding	132
	Pneumonia	133
	COVID-19	380
	Unknown	97

Table 3
Characteristic of dental subjects.

		Subjects
Sex	Female	242
	Male	205
Age	0–20	97
	21–40	209
	41–60	90
	61–80	49
	81–90	2

8. Conclusion

Many clinical AI algorithms have been proposed and achieved excellent performances in the clinical fields in the closed clinical and data island settings. In contrast, few clinical AI algorithms are applied in the real-world clinical setting. Clinical AIBench, as a benchmark suite or third-party tool, provides open and fair clinical settings for evaluating clinical AI algorithms. In addition, Clinical AIBench also provides a configurable clinical environment for developing clinical AI algorithms.

Clinical AIBench is an open and continuously evolving benchmark suite. Clinical AIBench will add the ICU and psychiatric disorders datasets soon.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Project of Guangxi Science and Technology, China (No. GuiKeAD20297004 to Y.H.), and the National Natural Science Foundation of China (No. 61967002 to S.T.).

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[3] S.G. Mueller, M.W. Weiner, L.J. Thal, R.C. Petersen, C.R. Jack, W. Jagust, J.Q. Trojanowski, A.W. Toga, L. Beckett, Ways toward an early diagnosis in alzheimer's disease: the alzheimer's disease neuroimaging initiative (adni), Alzheimer's Dement. 1 (1) (2005) 55–66.

[4] K. Tomczak, P. Czerwińska, M. Wiznerowicz, The cancer genome atlas (tcga): an immeasurable source of knowledge, Contemp. Oncol. 19 (1A) (2015) A68.

[5] S. Benjamens, P. Dhunnoo, B. Meskó, The state of artificial intelligence-based fda-approved medical devices and algorithms: an online database, NPJ Digit. Med. 3 (1) (2020) 1–8.

[6] P. Brocklehurst, D. Field, K. Greene, E. Juszczak, R. Keith, S. Kenyon, L. Linsell, C. Mabey, M. Newburn, R. Plachcinski, et al., Computerised interpretation of fetal heart rate during labour (infant): a randomised controlled trial, Lancet 389 (10080) (2017) 1719–1729.

[7] Y. Liang, Y. Guo, Y. Gong, C. Luo, J. Zhan, Y. Huang, Flbench: A benchmark suite for federated learning, in: BenchCouncil International Federated Intelligent Computing and Block Chain Conferences, Springer, 2020, pp. 166–176.

[8] W. Gao, F. Tang, J. Zhan, X. Wen, L. Wang, Z. Cao, C. Lan, C. Luo, X. Liu, Z. Jiang, Aibench scenario: Scenario-distilling ai benchmarking, in: 2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), IEEE, 2021, pp. 142–158.

[9] J. Zhan, Call for establishing benchmark science and engineering, BenchCouncil Trans. Benchmarks Stand. Eval. 1 (1) 100012.

[10] F. Zhang, C. Luo, C. Lan, J. Zhan, Benchmarking feature selection methods with different prediction models on large-scale healthcare event data, BenchCouncil Trans. Benchmarks Stand. Eval. 1 (1) (2021) 100004.

[11] C. He, S. Li, J. So, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, L. Shen, P. Zhao, Y. Kang, Y. Liu, R. Raskar, Q. Yang, M. Annavaram, S. Avestimehr, Fedml: A research library and benchmark for federated machine learning, arXiv preprint arXiv:2007.13518.

[12] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, Y. Khazaeni, Bayesian nonparametric federated learning of neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 7252–7261.

[13] C. Xie, K. Huang, P.-Y. Chen, B. Li, Dba: Distributed backdoor attacks against federated learning, in: International Conference on Learning Representations, 2019.

[14] S. Warnat-Herresthal, H. Schultze, K.L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N.A. Aziz, et al., Swarm learning for decentralized and confidential clinical machine learning, Nature 594 (7862) (2021) 265–270.

[15] Y. Huang, N. Wang, S. Tang, L. Ma, T. Hao, Z. Jiang, F. Zhang, G. Kang, X. Miao, X. Guan, et al., Openclinicalai: enabling ai to diagnose diseases in real-world clinical settings, arXiv preprint arXiv:2109.04004.

[16] Inspection, G.A.o.Q.S., Quarantine of the People's Republic of China, Standardization administration, http://std.samr.gov.cn//gb/search/gbdetailed?id=b691bb77876cd126e05397be0a0af3b3.

[17] S.G. Mueller, M.W. Weiner, L.J. Thal, R.C. Petersen, C. Jack, W. Jagust, J.Q. Trojanowski, A.W. Toga, L. Beckett, The alzheimer's disease neuroimaging initiative, Neuroimaging Clin. N. Am. 15 (4) (2005) 869.

[18] J.P. Cohen, P. Morrison, L. Dao, Covid-19 image data collection, arXiv:2003.11597. URL https://github.com/ieee8023/covid-chestxray-dataset.

[19] agchung, Figure1 Covid-19 Chest X-Ray Dataset Initiative, GitHub repository, URL https://github.com/agchung/Figure1-COVID-chestxray-dataset.

[20] p. agchung, lindawangg Linda Wang, Actualmed Covid-19 Chest X-Ray Dataset, GitHub repository, URL https://github.com/agchung/Actualmed-COVID-chestxray-dataset.